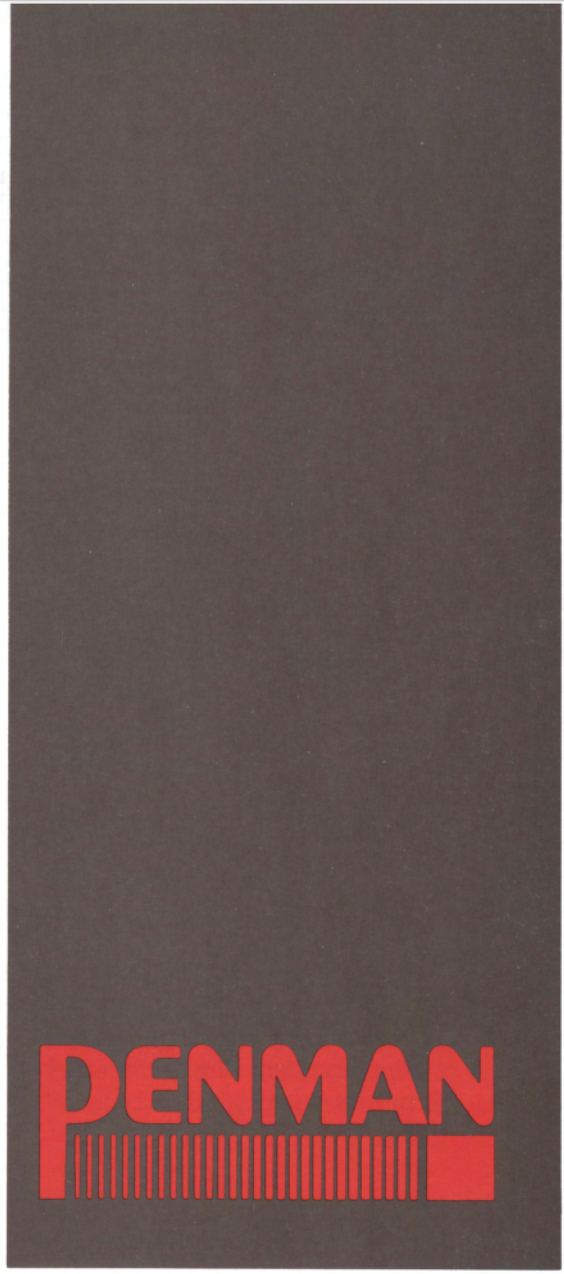
User Manual



User Manual

Penman Products reserves the right to make changes to any products described herein to improve their functioning or design.

Important note

This equipment generates and uses radio frequency energy, and if not installed and used properly, that is in strict accordance with the operating instructions, may cause interference to radio and television reception. It has been tested and found to provide reasonable protection against such interference. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, try to correct the interference by one or more of the following means:

- re-orient the receiving antenna
- relocate the equipment with respect to the receiver
- move the equipment away from the receiver
- plug the equipment into a different outlet so that the equipment and receiver are on different branch circuits
- if necessary consult your dealer or an experienced radio/television technician for additional suggestions.

Part number HA234538
First edition **September 1984**

Manual designed by Information Design, Trowbridge

Contents

For users

For programmers

For reference

Conventions

- shows a warning or important information

'Computer' means computer or terminal

Introducing Penman

Penman is a unique peripheral – part plotter, printer, robot, turtle and mouse.

You can use Penman to create colourful precision plots to illustrate your ideas.

In the office
In the office, you can use Penman to create organisation charts or to plot financial performance from spreadsheet data.

If you're a programmer
Logo or BASIC programmers can produce combined 3-colour drawings and procedure listings.

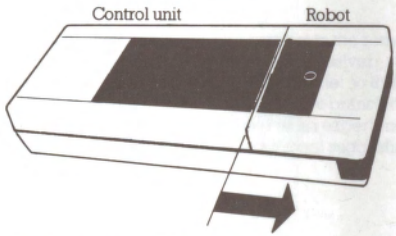
In the design studio
Graphic designers can use Penman for lettering large drawings, while home designers can produce outline drafts as templates for their hand-finished drawings.

At school or in the laboratory
You can plot experimental results, draw up wall posters, and experiment with robotics.

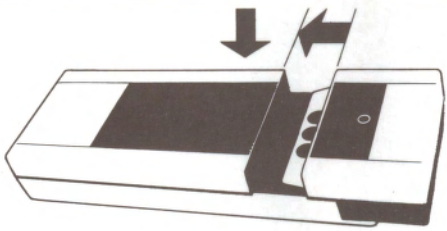
In short, you can use Penman wherever you need to put pen to paper!

Looking over Penman

Removing the robot
Gently but firmly pull the robot from the control unit. The ribbon can be withdrawn up to one metre from the control unit – do not pull out any further or pull harder than you need to. The ribbon is automatically clamped in position after being withdrawn. **Don't bend, twist or cut the ribbon!**

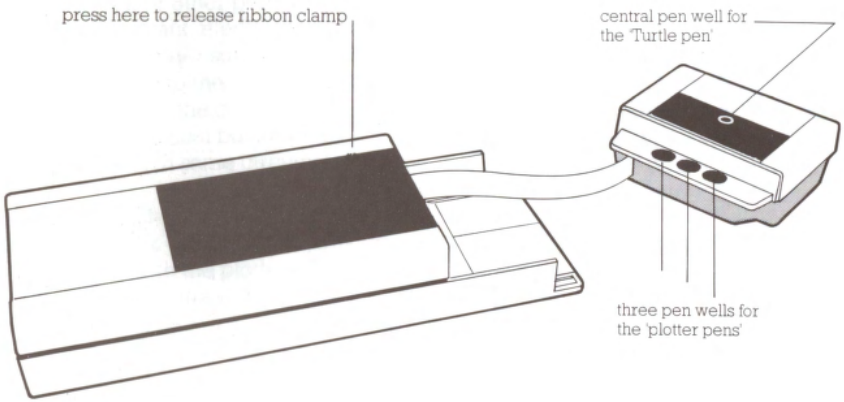


Replacing the robot
Press downwards firmly on the top right hand corner of the control unit.
Still pressing, hold the robot with your other hand and allow the ribbon to feed slowly back into the control unit *without kinking or twisting*.
Carefully slide the robot into the control unit, then release.

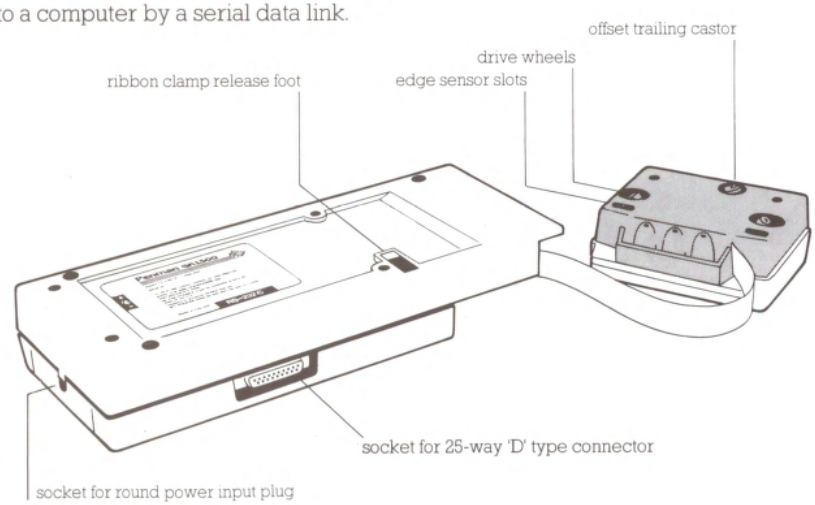


Don't try to dismantle the robot or control unit for any reason – it has no user-serviceable parts.

Internally the **robot** contains the servo motors, pen lift mechanism, opto-electronic sensors, and beeper.



Internally the **control unit** contains the circuitry to control the robot and interface to a computer by a serial data link.



How Penman works

Penman is an innovative design which uses micro electronics to reduce its mechanical complexity to just two moving parts – the robot's wheels (plus a simple pen lift mechanism).

The wheels are in fact the shafts of two motors. Penman works by controlling the relative speeds of the wheels, allowing the chosen pen to trace out any required path. This is how Penman is able to plot smooth rather than stepped curves.

While mechanically simple, Penman's electronics are very sophisticated. Its 8-bit microprocessor runs software which precisely models the action of the robot. Drawings to be plotted are broken down into individual lines, arcs or circles. The software then calculates the wheel motions needed to make the selected pen follow the lines, and sends instructions to the robot.

In the factory, Penman is automatically calibrated without mechanical or electrical adjustment by 'learning' about its environment.

In use, Penman maintains a 'model' of the robot's position in relation to the plotting paper edges. As it moves, the robot – like any mechanical device – is subject to minute drift, so that Penman's model and the robot's real position become progressively out of step. You can exactly realign robot position and Penman's model at any time by commanding the robot to 'home'. The robot then uses its opto-electronic sensors to align itself accurately with the paper edges.

In its 'robot' mode of operation, Penman's robot can be driven under direct control from a host computer – rather than by Penman command – across the serial link. When the robot is moved by hand, Penman sends back velocity and position information to the computer, enabling the robot to act as a 'mouse'.

Preparing to use Penman

Plotting surface

The robot needs a smooth, horizontal surface, sufficiently large to hold the paper with a 50mm (two inch) margin. The Penman Platen or other perspex sheet is ideal, because static electricity holds the paper flat. With other surfaces, lightly secure the paper to the plotting surface with sticky tape in the corners. Paper slippage is not critical but may cause inaccurate plots in some circumstances.

The plotting surface should be dark – preferably black – and matt, to provide good contrast with the plotting paper when the robot looks for the edge.

Paper

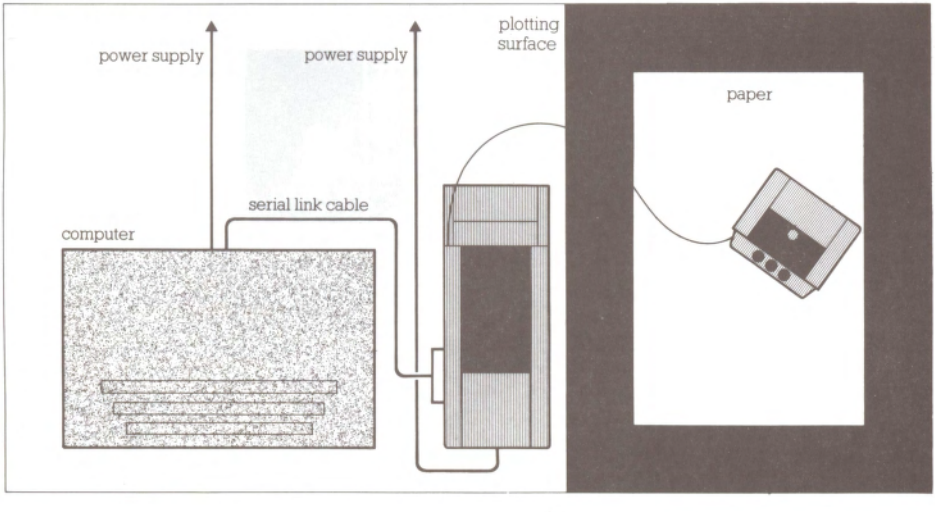
Recommended paper sizes are given below. Special applications can use larger paper, but you may then need to move Penman between plots.

On all paper sizes, allow a margin around the plotting area otherwise the wheels or castor of the robot may run over the edge. This is not serious but may cause misalignment in the plotted image.

The paper should be light in colour. For best results, use good quality paper at least 105gsm in weight.

Recommended paper sizes	
ISO	A4 210×297mm
	A3 297×420mm
ANSI	A 8½×11inches
	B 11×17inches

Suggested working layout



Power supply
Penman needs a standard power outlet for the power supply unit. Check that the operating voltage marked on your power supply unit matches your local electricity supply.

Serial link (connecting lead)
Penman needs a serial connector lead to connect with your computer. Leads are sold separately, for each make of computer. One end has a 25 way 'D' type plug, which fits into the matching socket on the side of the control unit. The other end of the serial lead has a connector for the computer serial port.

- *Don't let the serial and power leads tangle with each other or obstruct the plotting surface*

All communication between Penman's control unit and the computer is by means of the serial data link (RS232-C or equivalent). Penman can use three baud rates: 300, 1200 and 9600 baud – set your computer to 9600 baud if possible. The control unit automatically senses which rate is set. If your computer does not allow serial communications at one of these rates, consult your supplier.

Pens
The pen wells along the front of the robot take pens of the style supplied, or – with an adaptor placed in the well – ordinary Pentel SPC8 refills which you can buy in High Street stores.

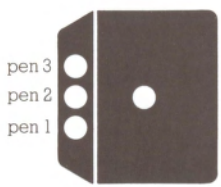
Pens are available with line thickness of 0.3mm or 0.7mm, and in a range of colours. Penman pens are similar to Hewlett Packard plotter pens, which you can also use.

Arrange the colours (and line thicknesses) as you require. Provided the pen wells are clean, the pens will just drop in and rest on the end of the pen lift arm (visible at the top of the pen well). If a pen does not fit or appears not to work in some way, **do not force it** – contact your supplier for advice or replacement.

The centre pen well is used for Turtle graphics, and takes Pentel refills only.

- *Remove pen caps before inserting the pens, and replace caps after use or during any long intervals between plotting*

- *Don't use the centre pen at the same time as front pen 2.*



Using Penman

Each time you set up Penman for use, carry out the 'self-test' routine and produce the test plot before you connect Penman to the computer. See the page **Self-test routine**.

When you have self-tested Penman, connect it to your computer with the serial link cable.

- *If you meet errors while using Penman, see the page* **What to do if things go wrong**

You can control the robot in several ways. You will need one of the following:

The Penman utility cassette or disk for your computer. This includes the program PENTLK for sending commands to Penman, together with other utilities for using Penman
▶ Run **WELCOME**

An *application program* with drivers for Penman built-in
▶ read documentation for application program

A *language system* (eg Logo or BASIC) which has drivers for Penman built-in
▶ read documentation for language

A *terminal emulator* program for your computer, which should be able to send direct commands to Penman
▶ green section and carry out the **Introductory tour**; read terminal emulator documentation

---

You can also use Penman if you have a good working knowledge of your computer's operating system and a suitable programming language. The operating system should include utilities to send ASCII strings over a serial port, either directly or from a program. Use these utilities to command Penman.
▶ green section and carry out the **Introductory tour**; read the operating system manual.

Introductory tour

Before using the commands given later in this section, try out the command sequences shown below in green letters. These demonstrate the main Penman commands, and show how easy it is to control the robot.

You will need a terminal emulator program or PENTLK running in your computer. Use A4 paper ('A' size in US), with the robot placed as shown in 'starting position'.

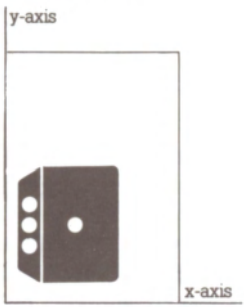
1 *Initialise and 'home'*

Type **I**

• If you are controlling Penman from an operating system or language system, *send* these commands rather than type them. For example, in BASIC `PRINT "I"`

Penman returns `Penman version n.n` to your screen.
(If Penman was already switched on and in use, the **I** command initialises Penman instead – see the next command below.)

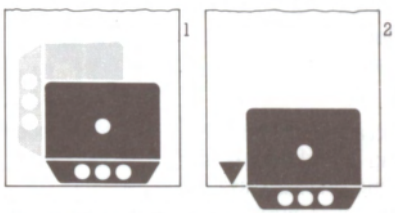
Starting position



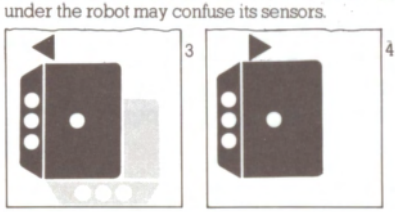
If the robot doesn't respond, check that

- the serial link is correctly fitted and enabled from the computer
- the computer is transmitting at 300, 1200 or 9600 baud
- the power supply is correctly connected and turned on
- If all seems correct, switch off the power to Penman, switch on, and try step 1 again. If there is still no response contact your suppler for advice.

Homing sequence



If the robot doesn't 'recognise' the edge, try a darker surface under the paper. NB Very bright light getting under the robot may confuse its sensors.



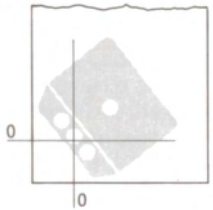
Type **I** again
This resets Penman to its 'initial' state.

Type **H**
Penman goes through the four steps shown to find the corner – it 'homes'.

After homing, Penman knows the position of the corner of the page. Penman's origin is 50mm from each edge. The robot comes to rest with pen 2 at co-ordinates −20, +25 mm from the origin.

2 *Move pen 2 to the origin*

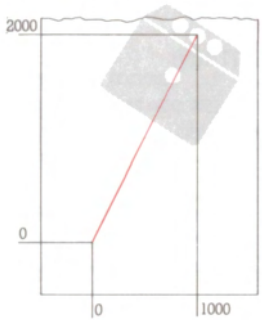
```
U A M 0,0,
```



`U` pen up
`A M` absolute cartesian move
`0,0,` to origin (0,0)

3 *Draw a line from origin to (1000,2000)*

```
D A M 1000,2000,
```



`D` pen down
`A M` absolute cartesian move
`1000,2000,` to the point with these co-ordinates

Note how the robot moves in a curve while drawing a straight line with pen 2.

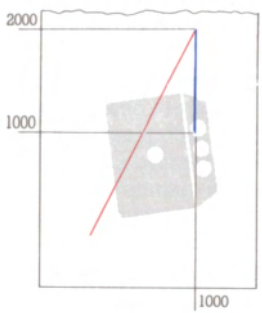
4 *Select a new pen*

```
P 1,
```

The robot rotates to move pen 1 over the position last occupied by pen 2.

5 *Draw a line using a relative move downwards*

```
R M 0,-1000,
```



`R M` relative cartesian move
`0,-1000,` by these units, with respect to the robot's current position

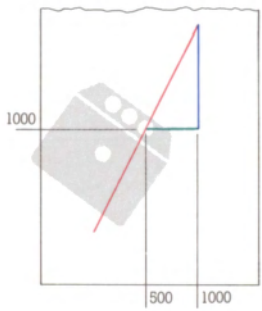
Penman remembers (from step 3) that the pen is down.

6 *Complete a 3-colour triangle*

For the third colour, select pen 3:

```
P3,
-500,0,
```

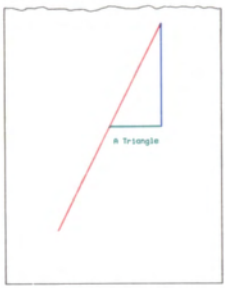
Penman remembers the *mode definitions* from previous steps (pen down, relative move).



7 *Print some text*

```
U 0,-160,
L ATriangle ↩
```

(↩ means press RETURN)



8 *Home again*

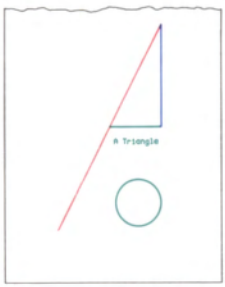
```
H
```

The robot re-checks the position of the corner, improving its future accuracy. In the process, it unwraps its cable.

9 *Draw a circle*

```
A M 1000,200,
D V $1000,$4000,$10000,
```

With pen up, the robot moves to (1000,200) and then draws a circle of radius 22.5mm (see **Polar commands** for explanation).



As an exercise, first send Penman home, and then label your circle

More about commands

Penman has three types of commands:

- Moving and plotting commands
- Character and text commands
- Utilities

Moving and plotting commands
The same commands cover both moving and plotting – the difference lies in whether the selected pen is up or down.

With *moving*, the selected pen is in the up position and makes no mark on the paper while the robot moves the pen to the specified position.

In *plotting* the selected pen is in the down position, and draws on the paper while the robot moves.

To plot, you can select any of the three front pens by number. This effectively enables you to select colour and line thickness by command, if the required pens are placed in the wells.

The robot performs three kinds of movement, with a command for each:

absolute cartesian – movement *to* a set of co-ordinates with respect to the origin

relative cartesian – movement *by* a set of co-ordinates with respect to the pen's current position

polar – movement by vector from the pen's current position, resulting in smooth arcs, circles (and straight lines if required)

Each of these movements is explained in the appropriate command description.

Character and text commands allow you to label plots or print in letter sizes from 1mm to 127mm high (1/25inch to 4 inches) in a variety of slants (italics) and orientations.

Utilities include initialising, examining memory and robotics commands. In robotics mode you can control the robot directly at a low level, and examine the movements of the wheels (for applications where the robot is moved manually).

Penman commands can also be used to achieve *turtle graphics* movements and plotting – see the examples on the last page of the green section.

All of the commands are fully explained in **Using commands** and the following green pages. The **Command summary** is for quick reference.

Command format
All commands are single characters. They are usually printable characters, although a few are not.

Some commands require further information to be supplied with them, as *parameters*. Parameters are numbers in decimal or hexadecimal form.

Other commands set a *mode*, so that you can then send a stream of data to Penman and have Penman deal with it appropriately. When in label mode for example, Penman prints whatever ASCII characters you send it until you turn label mode off.

How Penman deals with commands
Penman's control unit accepts commands from the computer over the serial link.

Penman sends a prompt to the computer when it is ready to accept a command. When the computer sends a command, either from a program or directly from the keyboard, Penman checks the command syntax and makes sure that any command parameters are in the correct range. If a command is incorrect, Penman beeps and sends an error prompt to the computer.

□ When a command is accepted and completed, Penman sends a carriage return character, line feed character and an exclamation mark.

□ When Penman rejects or abandons a command (for example if an obstacle prevents completion of a movement) it sends a
question mark, a BEL (beep) character and an exclamation mark.

• *Penman doesn't send prompts while it is in robotics mode.*

When busy, Penman sends a 'handshake' signal to the computer to hold up further commands. If the computer continues sending commands, Penman sends an error prompt and the robot gives a continuous beep.

In the following pages, each command title is followed by the command *syntax*, for example

```
V $distance,$direction,$curvature
```

Commands are shown in capitals, with parameters in small letters. (You can *enter* commands in capitals, small letters or both.) Parameters which must be in hexadecimal have $ in front.

Put $ in front of all hexadecimal numbers. Decimal numbers (only) may have a – or + sign in front of them. (Use two's complement form for negative hexadecimal numbers.) Decimal points are not allowed.

Separate all commands and parameters either by a space or a comma. Our examples show commands separated by spaces and parameters separated by commas – for example

```
R M 0,0,
```

`R` and `M` are commands. `0,0` are `M`'s parameters. Commands with parameters end with a space or comma too. Commands without parameters need no terminator at all.

Commands can be strung together, separated either by spaces or commas. You can enter a ↩ after each command or command sequence, in place of a comma or space. For example

```
R M 0,0 ↩
V $1000,$400 ↩
```

The second example shows how this is used in the two-parameter version of a polar move where the third (optional) parameter is omitted. You cannot use ↩ in between mandatory parameters.

- The carriage return character ↩ means <CR> and not <CR><LF>. If <LF> is sent to Penman while in command mode, it is ignored.

For further examples of commands and command sequences, see the **Introductory Tour**.

- *Each time you start a new plot or move the robot manually, initialise Penman's internal X-Y and orientation variables – see the page* **Initialising Penman**

Initial state
When you switch Penman on, it assumes values or settings for certain commands and parameters. We show these in command entries under the heading *initial state*.

Symbols and conventions

↩ means press RETURN or send a carriage return character

Hexadecimal numbers have $ in front – for example $400

ASCII character representations are in decimal – for example ASCII 16 (= ASCII $10)

BAM means Binary Angular Measure, whereby $10 000 = 360°

Pen selection

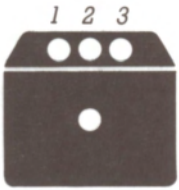
Set pen number

syntax
P1, selects pen1
P2, selects pen2
P3, selects pen3

initial state pen 2

Selects the pen to be used for plotting with subsequent commands – pen 1, pen 2 or pen 3. For the centre pen, select pen 2 (the initial state).

The actual pen type, colour or line thickness depends on the pen placed in the chosen pen well. You can thus use three colours straight away, with more if you change pens during a plot.



Pen up

syntax **U**
initial state up

Immediately lifts the currently selected pen and leaves it in the up position for further moves. No further lines will be drawn until you select pen down with command **D** (or until you select label mode with **L**, or robotics mode with #).

Pen down

syntax **D**

Lowers the current pen on to the paper when Penman receives the next move command. The pen is not lowered immediately.

If the pen is down and no valid move command or data string is received for 0.5 second, Penman automatically lifts the pen to prevent ink blots. When it receives a move command Penman lowers the pen again before it moves off. To change the auto penlift delay, use the C command to alter the PENTIME variable: $10=1sec, $08=0.5sec.

Cartesian moves/plots

Moving or plotting in straight lines between X–Y co-ordinates

Set absolute move

syntax `A`
initial state absolute mode

Sets Penman in absolute mode, so that subsequent cartesian moves send the robot to co-ordinates with respect to *the origin*.

The origin is defined by the 'home' command **H**, initialisation command **I**, or origin change command **O**.

Set relative move

syntax `R`

Sets Penman in relative mode, so that subsequent cartesian commands move the pen *by* co-ordinates relative to *the pen's current position*.

General move/plot

syntax
`M x co-ordinate,y co-ordinate,`
sets Penman in cartesian mode, then moves or plots *to* (absolute) or *by* (relative) the specified co-ordinates
initial state cartesian mode

This command covers both moving and plotting – if the currently selected pen is up, the robot merely moves; if the pen is down, it plots. The exact movement depends on whether Penman is in absolute mode or relative mode (see the **A** and **R** commands).

M also sets Penman in cartesian mode. This takes Penman out of polar mode, if previously set. You can then send Penman a series of X/Y co-ordinates which it interprets according to its absolute or relative mode, moving the robot appropriately.

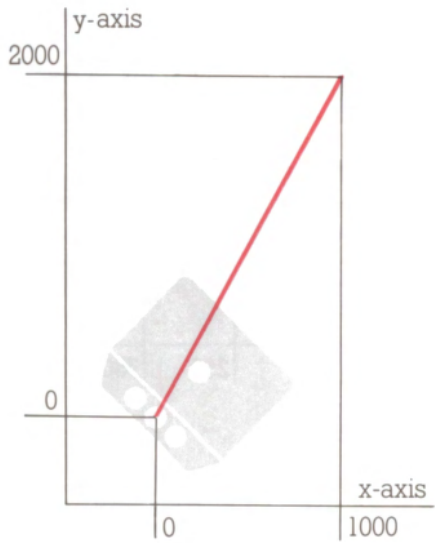
x co-ordinate and **y co-ordinate** can be:

□ **signed decimal integers** in the range +8191 to −8191. Each decimal unit represents **0.1mm real scaling**. Decimal points are not allowed.

□ two's complement **hexadecimal numeric strings** in the range $0 to $FFFF. Each hexadecimal unit represents **0.025mm real scaling**, giving greater control over the robot's movements than with decimal co-ordinates.

• *In cartesian mode Penman continuously updates two internal variables holding the current absolute X and Y co-ordinates, even when the robot meets an obstruction (but not if you move the robot manually). After an obstruction the robot may have an uncertainty of position of ≃ 1mm.*

Example absolute cartesian move
The robot is shown with pen 2 over the origin (0,0 absolute co-ordinates).



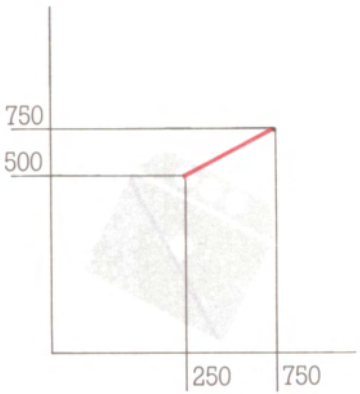
To move pen 2 to the point (1000,2000):

```
A M 1000,2000,
```

If cartesian mode has been previously selected you can leave out the **M**. If absolute mode has been selected too, you can leave out both the **A** and the **M**, and supply just the co-ordinates.

Example relative cartesian move
In relative cartesian mode all movement is relative to the *current position* of the selected pen. For example, the command sequence **R M 0,0,** causes no move, whereas the absolute equivalent **A M 0,0,** causes the pen to move to the origin (0,0).

If the robot is in the position shown below with the selected pen at the point 250,500 (absolute) the sequence **R M 500,250,** causes the pen to move 500 units along the X axis and 250 units up the Y axis from its current position, as shown by the red line.



• *In relative mode any command sequence always produces identical results, whatever the robot's current position (paper permitting). In absolute mode you must calculate fresh co-ordinates each time.*

Polar moves/plots

Drawing smooth arcs, circles and lines

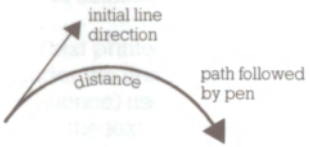
Polar (vector) move

syntax

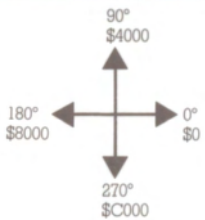
`V $distance,$direction` ↩
sets Penman in polar mode, then draws a straight line

`V $distance,$direction,$curvature,`
sets Penman in polar mode, then draws an arc or circle

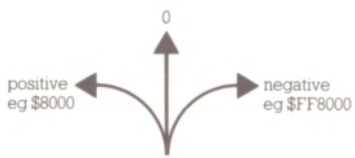
Once a V command has been issued, Penman interprets subsequent hexadecimal strings as polar move commands until polar mode is overridden



One unit of **distance** is 0.035mm, given by ($\pi \times 45$)/4096 mm



Direction is that of the selected pen as it moves off, measured in BAMs



Curvature is a *two's complement* number with up to 6 hexdecimal digits, given by ($8000×45)/radius in mm.

The command

`V $distance,$direction,$curvature,`

causes Penman to move off initially with the selected pen following the direction given. For an arc or circle, the heading direction changes as soon as Penman starts moving. The result is a smooth curve, starting from the pen's original position and proceeding for the specified distance as measured *along the curve*.

The relationship between distance and curvature is

distance × curvature = angle subtended by arc at centre of curve

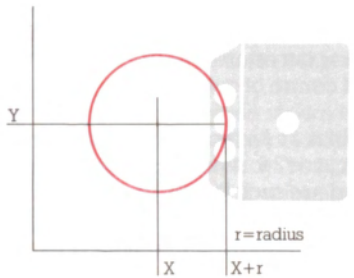
The angle is 2π for a complete circle, π for a semi-circle, $\pi/2$ for a quarter circle. The units used are such that 2π is expressed as $10000000.

Example
The command
`D V $1000,$4000,$10000`
plots a *complete* circle, because $1000×$10000 is $10000000. The circle's radius is ($8000×45)/$10000 mm – that is, 22.5mm.

Penman's initial heading is $4000 – along the y-axis. The curvature is positive, so it will bear off toward the left (negative x-axis). The centre of the circle will be (−½×450,0) from the initial position of the selected pen.

Drawing circles

To draw a circle of known radius and centre, compute a convenient point on the circumference – for example, offset along the x-axis as shown. Use an absolute cartesian command to move the pen to (X+r, Y). Now use a polar move to draw the circle, bringing the pen back to its starting position: **direction** is $4000 (parallel with the y-axis); **curvature** is ($8000×45)/radius in mm; **distance** is $10 000 000/curvature.



Drawing a straight line

Enter only the **$distance** and **$direction** parameters, and end the **V** command with ↩.

• The V command **does not** update Penman's internal X, Y coordinate register, so subsequent absolute cartesian commands will not work properly unless you update them from the computer using the O command. (In the special case of drawing a circle, the pen returns to its starting point – here the X, Y coordinate register will be correct.)

Character & text plots

Printing text, or adding captions to plotted drawings. Use a cartesian or polar move to position the pen where the text is to begin.

Label (text)

syntax `Ltext`

Puts Penman in text mode – all subsequent ASCII characters sent to Penman are printed as text according to current values for size (**S**), orientation (**Q**), and slant (**Z**).

While in text mode the robot takes no action until it receives a printable character – see **Character set table** for the full Penman character set. Penman ignores unrecognised characters.

□ to *terminate text mode*, use ↩. This returns Penman to command mode.

□ for *new lines* (text printed on the next 'line' starting under the first character in the present text sequence) use <LF> (ASCII 10) at any point in the text string. This carries out the normal ↩,<LF> sequence.

□ to *backspace* the text plotting position by one character, use <BS> (ASCII 8).

Character size

syntax `Ssize`
initial state `S5`

Sets the size in millimetres, from 1 to 127 decimal, to be used in plotting further text.

Text with S5

Text orientation

syntax `Qorientation`
initial state `Q0` (x-axis)

Sets the quadrant used for subsequent text plotting – Q0, Q1, Q2, or Q3. The four orientations are:

Character slant

syntax `Zslant`
initial state `Z0` (upright text)

Sets the slant (forward or backward italic) for subsequent text plots. The range is +9 to -9. Use positive numbers (+ sign optional) for right-slanting text, negative numbers for left-slanting text. The greater the number, the greater the slant: you'll probably find the range +3 to -3 most useful. The *angle* of slant is given by arctan (Z/4).

Text with Z+1

Text with Z-1

Initialise

syntax `I`

Sets initial values for the robot's X-Y co-ordinates and angular rotation (THETA), and selects pen 2 in the up position. Use **I** when you manually reposition the robot, or after the self-test routine when setting Penman up for use.

Variables reset
XCOR → −200 decimal units
YCOR → +250 decimal units
THETA → $4000 BAMs
PENST → pen 2, up position

Home (set datum)

syntax `H`

Returns the robot to the 'home' position (if not already there), re-aligns it with both edges of the plotting paper, and selects pen 2 in the up position. This resets Penman's coordinate system relative to two edges of the paper.

If the robot cannot distinguish the paper edges after moving 160mm along any axis, or if it meets an obstruction, Penman abandons the command and sends an error prompt to the computer. It updates the internal X-Y co-ordinates nonetheless.

Variables reset
XCOR → −200 decimal units
YCOR → +250 decimal units*
THETA → $4000 BAMs
PENST → pen 2, up position
*YCOR may differ by a small amount, but will correctly reflect pen position.

Align with edge

syntax `E`

Moves the robot slowly forward for up to 160mm in search of an edge of the plotting surface (or a white-to-black surface change). If it detects no edge within 160mm or meets an obstruction, Penman abandons the command and sends an error prompt to the computer. In either case X, Y and THETA are updated as for a normal move. This physically aligns the robot with an edge but does not reset the coordinate system.

After an E command, the front pens are 300 decimal units over the edge of the paper, and the robot is precisely in line with the paper edge.

Beep

syntax `B`

Robot emits a sound for 0.5 secs.

Set/reset auto unwrap

syntax
`W1,`
enables auto-unwrap
`W0,`
disables auto-unwrap

initial state auto-unwrap disabled

The auto-unwrap facility prevents the robot from becoming entangled in its connecting ribbon.

When you enable (turn on) auto-unwrap, Penman checks the state of the internal variables WRAP and THETA. If these exceed 0,$DFFF or are less than -1,$9000 the robot rotates on the spot to free the ribbon. Penman repeats the check after each subsequent command until you turn auto-unwrap off.

Set new internal X and Y co-ordinates

syntax `O newXCOR,newYCOR,`

This allows you to alter the robot's current internal X and Y co-ordinates, effectively changing the origin. Parameters for the X and Y co-ordinates must be in X-Y order, and can be:

□ signed decimal integers in the range +8191 to -8191. Each decimal unit represents 0.1mm real scaling. Decimal points are not allowed.

□ two's complement hexadecimal numeric strings in the range $0 to $FFFF. Each hexadecimal unit represents 0.025mm real scaling.

Examine/change variables

syntax
`C $address` ↩
examine variable
`C $address,$newvalue,`
alter variable

Use **C** to examine Penman's internal variables (registers) and if necessary change any values found – see **Useful variables**.

`address` is up to two hexadecimal digits, and refers to one byte of Penman's internal variables.

`newvalue` Penman displays the current value for the address. Enter the new value as a one or two digit hexadecimal number and then press ↩

Reset

syntax <escape> ASCII 27

Resets Penman to its power-up state. This command is executed immediately.

Change optical sensor levels

syntax
`X2,`
resets edge sensors for new paper/background combinations
`X1,`
takes one edge sensor reading for use as a comparison level
`X0,`
re-calibrates the wheel encoders

Penman uses four optical sensors to encode the wheel rotations as it moves, and two more to align with the paper's edge.

The *wheel sensors* monitor slotted disc encoders, providing Penman with the precise control necessary for plotting.

Penman compares the voltage signal from each *edge sensor* with a value held in memory, and decides whether the sensor is looking at light or dark. (This is how the edge sensors tell Penman what is paper and what is not.)

Each time you switch Penman on, it reads the values from a special store which was set at the factory when Penman was 'calibrated'.

■ To set new values to allow for different shades of paper and background, use the commands **X2** and **X1**:

X2 re-calibrates the edge sensors for new paper/background combinations. Put the robot in the self-test position and enter `X2`. The robot looks at the paper, moves forward and looks at the background, and

then calculates an in-between value for each sensor to use on subsequent Home commands (**H**).

X1 looks through the sensors and holds the measured values.
This is similar to **X2**, but the robot takes only a single reading and uses this as a comparison level.

■ To re-calibrate the wheel encoders, replace the robot in the control unit and enter **XØ**. Both wheels spin at high speed, correctly setting the comparison values for the four optical sensors (two for each wheel).

• *Make sure that the wheels are unobstructed before using* **XØ**, *or Penman may not work properly afterwards.*

• *The changes you make with this command are temporary. The next time you switch Penman on, the factory-set values will be re-instated.*

Enter robotics mode

syntax #

Exit robotics mode

syntax <DLE> ASCII 16

In robotics mode all data sent to the plotter is treated as packed binary data of which the lower 7 bits are interpreted as follows:

bit 6 - pen 2 control
0 pen 2 up
1 pen 2 down

bits 5,4 - general motor control
00 switch off motors (disable servos)
01 not used
10 enable servo control and hold position *with* deadband
11 enable servo control and hold position *without* deadband

bits 3,2 - left motor control
00 no change
01 increment demand position *clockwise*
10 increment demand position *anti-clockwise*
11 set positional error to zero

bits 0,1 - right motor control
As for left motor control, above

• *Any eighth bit transmitted will be masked out/ignored*
• *Note the disallowed state of bits 5,4 above. It is not possible to control pens 1 and 3 in robotics mode. The wheels are controlled by single increments in rotation, either clockwise or anti-clockwise, corresponding to a movement of ~0.07mm. Positive rotation of the wheels is defined as clockwise when viewed from under the robot. No error prompts are echoed to the host if an obstruction is met, but the positional error information (see below) will reflect the situation correctly.*

In robotics mode the following characters produce one increment of movement as described:

ASCII 9	robot moves forward
ASCII 6	robot moves backward
ASCII 1	robot rotates anticlockwise about left wheel
ASCII 2	robot rotates clockwise about left wheel
ASCII 4	robot rotates anticlockwise about right wheel
ASCII 8	robot rotates clockwise about right wheel

While in robotics mode Penman returns useful information after each command. The data consists of 8 bits, sent back as two ASCII hexadecimal characters. You can interpret the returned data byte (two ASCII characters) as follows:

bits 7,6,5
left wheel positional error (in steps)

bits 4,3,2
right wheel positional error (in steps)

bit 1
left paper edge sensor
1 white, **0** black

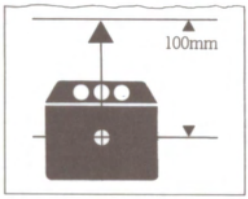
bit 0
right paper edge sensor
1 white, **0** black

The servos internally accomodate a positional error of 31 steps before automatically disengaging. The servo control is of second order type exhibiting velocity lag, so returned positional data will usually contain a positional error.

You can produce 3-colour plots from Logo by using the 'plotter' (front) pens and the direct commands on pages 15 to 25. But if you want to make Penman *behave* as a traditional Logo turtle, using its central pen well to leave a trail, use the command sequences shown below.

Forward 100mm

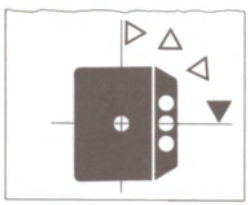
```
I D V $B54,$8000,$0,
```



To move forward 1mm
`I D V $1D,$8000,$0,`
To move forward 10mm
`I D V $122,$8000,$0,`
To move *backwards* 100mm
`I D V $B54,$0,$0,`

Right 90

```
I V $800,$4000,$FF8000,
```



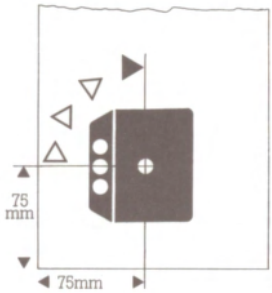
To turn right(approximately) 1 degree
`I V $17,$4000,$FF8000,`
To turn right(approximately) 10 degrees
`I V $E4,$4000,$FF8000,`
To turn right 45 degrees
`I V $400,$4000,$FF8000,`
To turn right 180 degrees
`I V $1000,$4000,$FF8000`
To turn *left* 90 degrees
`I V $800,$C000,$8000,`

Aligning Turtle heading

Place robot near lower left corner, facing left edge of paper.

```
I W0, H V $800,$4000,$FF8000,
```

This rotates the 'turtle' through 90 degrees



• *Remove any pen in pen-well 2 before using the centre pen-well.*

For the advanced user

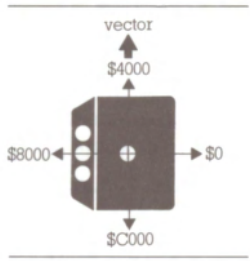
In the examples above, each command begins with **I** so the move is *relative to the turtle*, not relative to the paper. Using **I** avoids having to know which way the turtle is facing, but you cannot use the 'home' and 'unwrap' facilities.

If the computer is keeping track of the turtle's orientation (for example, if there is a screen turtle under its control) you can use another form of command which allows 'home' and 'unwrap'.

Penman stores the robot's orientation as THETA, which you can examine or change with the **C** command. (Addresses $89 (MSB) and $8A (LSB).)

After an I command, THETA is always $4000. After any other command, THETA is updated automatically. The computer program can either interrogate Penman to find THETA, or model it and keep a running value itself.

The robot's heading does not correspond to the traditional turtle heading. By our convention, THETA is the vector lying along the wheel axis.



Character set

A	65	a	97	0	48		32
B	66	b	98	1	49	!	33
C	67	c	99	2	50	"	34
D	68	d	100	3	51	#	35
E	69	e	101	4	52	$	36
F	70	f	102	5	53	%	37
G	71	g	103	6	54	&	38
H	72	h	104	7	55	'	39
I	73	i	105	8	56	(	40
J	74	j	106	9	57	)	41
K	75	k	107	:	58	*	42
L	76	l	108	;	59	+	43
M	77	m	109	<	60	,	44
N	78	n	110	=	61	-	45
O	79	o	111	>	62	.	46
P	80	p	112	?	63	/	47
Q	81	q	113	@	64		
R	82	r	114				
S	83	s	115				
T	84	t	116				
U	85	u	117			[	91
V	86	v	118			\	92
W	87	w	119	{	123	]	93
X	88	x	120	¦	124	^	94
Y	89	y	121	}	125	_	95
Z	90	z	122	~	126	£	96

Useful variables

for Penman version 2.1

name	address	bytes	description
XCOR	$89	2	Current position of selected pen in absolute coordinates* (units 0.025mm)
YCOR	$8B	2	Current position of selected pen in absolute coordinates* (units 0.025mm)
THETA	$87	2	Angle of wheel axis with absolute X axis in BAMs
WRAP	$8F	1	Overflow or underflow from THETA, representing the number of complete revolutions made by robot in either direction
PENST	$97	1	Controls current pen. *Bit 7:* 0 pen up, 1 pen down. *Bits 0,1:* current pen number
PENTIME	$8E	1	Controls the timer for auto penlift. $08 keeps the pen down for 0.5 second, $10 for 1 second and so on
LPOS	$9E	1	Left wheel position. *Bits 2–7:* modulo 32 count, units 0.07mm. *Bits 0,1:* encoder state. Used in mouse applications
LVEL	$A0	1	Left wheel velocity. Signed values positive means clockwise. Max range of values is ±280mm/sec. Used in mouse applications
RPOS	$9F	1	Right wheel position (as for LPOS above). Used for detecting change of position
RVEL	$A1	1	Right wheel velocity (as for LVEL above)
EDGER	$A7	1	Value for white/black threshold on right hand edge sensor. May be varied to adjust sensitivity
EDGEL	$A8	1	Value for white/black threshold on left hand edge sensor. May be varied to adjust sensitivity
CMDFLAGS	$BF	1	Indicates present state of Penman. *Bit 7:* wrap enabled. *Bit 3:* 0 relative, 1 absolute. *Bit 2:* 0 M, 1 V
SRVOFL	$A2	1	*Bits 4,3:* 01 servo enabled, 10 servo disabled *Bit 0:* 1 deadband enabled, 0 deadband disabled.
	$10	1	Serial rate & mode register**
	$11	1	Serial control and status register**. Normally $3A – if you set it to $18, Penman no longer sends responses to the computer
	$12	1	Serial data in register**
	$13	1	Serial data out (transmit) register**

*Polar moves cause errors in these values
**see Motorola 6801 data sheet

Serial link

Penman can deal with any of 300, 1200 or 9600 baud. It needs one start bit, eight data bits and one stop bit for correct decoding.

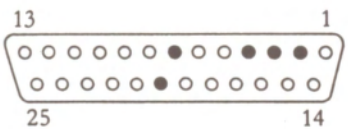
Penman uses *handshaking* to prevent the computer sending any more data when Penman is busy.

Handshaking can be either hardwired (via CTS or DTR) or soft protocol (using <TXON>, ASCII 17, and <TXOFF>, ASCII 19). The computer must be able to stop transmission within four characters of receiving either of these signals. If more than four characters are received while DTR is high (or before Penman sends TXON) the commands are corrupted. Penman detects this and sends a continuous beep until reset (powered off and on).

Penman always generates responses to commands. You can prevent the responses being sent back to the computer with the command **C $11,$18**. Responses will then be lost. To allow responses again, use the command **C $11,$BA**.

Wiring Connections of the 25-way 'D' connector

pin	name	comment
2	DATA FROM PLOTTER	data in to host system
3	DATA TO PLOTTER	data out from host system
4	CTS	hardwired to pin 20 internally
7	SIGNAL GROUND	
20	DTR	busy line from Penman (pin 4 also) (Logic high=true)



Command summary

Pen selection

Cartesian moves/plots

Polar vector moves/plots

Character & text plots

Utilities

P Set pen number	Choose current pen (1-3)
U Pen up	Lift pen immediately & leave up
D Pen down	Put current pen down on next move
A Set absolute mode	Subsequent moves are *to* absolute coordinates (0.1mm)
R Set relative mode	Subsequent moves *by* relative coordinates (0.1mm)
M General move/plot	Move by/to specified coordinates
V Polar move/plot	Move in lines and smooth curves using polar coordinates and curvature parameter (0.03mm)
L Label (text)	Plot following ASCII data as text plots
S Character size	Set current text characater size (mm)
Q Text orientation	Set axis along which text is plotted
Z Character slant	Set +/– slant of characters (italics)
ASCII 8 <BS> Back space	Backspace one character width
ASCII 10 <LF> Line feed	Equivalent to <LF,CR> on printer
ASCII 13 <CR> Text terminator	Terminate text mode
I Initialise	Reset THETA, XCOR, YCOR & PENST to default
H Home (set datum)	Check/reset origin to corner of paper
E Align with edge	Align with edge/datum line (forwards)
B Beep	Sound audible warning
W Set/reset auto-unwrap	Enable wrap state check and action
O Set new XCOR & YCOR	Set plotter internal X and Y coordinates
C Examine/change variables	See and change plotter RAM variables
ASCII 27 or <esc> Reset	Penman system reset – equivalent to power on
X Change optical sensor levels	Reset edge sensors for new paper/background combinations; recalibrate wheel encoders
Robotics	
# Enter robotics mode	Gives direct hardware control
ASCII 16 <DLE> Exit robotics mode	Return to Penman command mode
$ Hex operator	Hexadecimal parameter follows
, comma	Command/parameter separator
ASCII 32 <spc> space	Command/parameter separator
ASCII 13 <CR> carriage return	Command/parameter *terminator*

Self-test routine

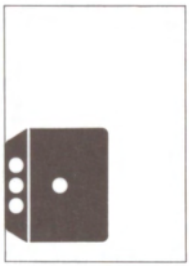
Use Penman's self-test routine to check that the unit is working correctly before you start plotting. You should not connect Penman to the computer to do this.

1

Withdraw the robot fully from the control unit and place it on a sheet of paper.

2

Place the front edge of the robot exactly as shown in the diagram.



Self-test starting position

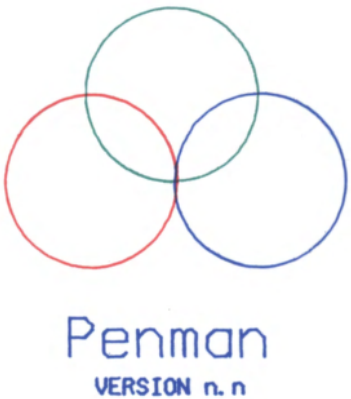
3

Place three pens or Pentel refills with adaptors in the front pen wells.

4

Connect the power supply cable to the round power socket at the end of the control unit, and switch the power on. The plotter should:

- move forward about 40mm (1.5 inches)
- move backwards about 40mm (1.5 inches)
- plot the message 'Penman version n.n' and draw three circles in the three pen colours.



Self-test plot

If Penman moves forward and back but then stops, the edge sensors cannot detect the paper edge. Check that

- the paper is white/light in colour and the background is black/dark.
- the edge sensors are not obscured
- no intense light is getting under the robot

If Penman doesn't produce the test pattern properly, check that

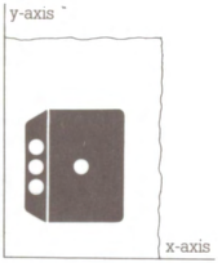
- the serial lead is disconnected
- the robot is correctly positioned with the front of the pen wells over the edge of the paper
- pens are inserted and can move freely
- the power supply is correctly turned on

If the plotter still fails to work properly, contact your supplier for advice.

Initialising Penman

Before initialising

- Connect Penman's power and serial leads
- Withdraw the robot from the control unit to the full extent of the ribbon and place it on the paper. Exact positioning is not critical but should be close to that shown.
- Switch on the power supply to Penman.
- Enable the serial port on the computer for transmission/reception (if in doubt, consult your system manual). The best serial communication rate for the plotter is 9600 baud – set it to this rate if possible.



Initialising Penman

1

Send an ASCII `I` (chr$73) or <carriage return> (chr$13) to Penman. (If in terminal mode on the computer, you'll see Penman's response `Penman version n.n` It sends this when it correctly interprets the baud rate).

2

Wait a second to allow hardware/firmware initialization, then send an ASCII `I` (chr$73) character to Penman to say that you have placed it in the initialisation position.

3

Send an ASCII `H` (chr$72) character to Penman. The robot now moves and, using its edge sensors, aligns itself with the two adjacent edges of the corner. It is then in the 'home' position – the internal X coordinate is −200 decimal units, the internal Y coordinate is +250 decimal units, THETA is set to $4000 BAMs, and pen 2 is selected, in the up position.

The plotter is now initialised and ready to receive commands. Each time you start a new plot or move the robot manually, re-initialise the plotter.

If the robot doesn't align itself properly, check that

- the serial link is correctly fitted and enabled from the host
- the host is transmitting at 300, 1200 or 9600 baud
- the power supply is correctly connected and turned on
- If all seems to be correct, switch off the power to the plotter, switch on again, and re-initialise as shown above. If there is still no response contact your suppler for advice.

For the advanced user

The above starting-position for the robot is recommended, but you can initialise Penman with the robot in any position on the plotting surface. If you do so, ribbon auto-unwrap may not work completely and the home command **H** may appear to malfunction.

Paper sizes & distance units

The table below shows recommended paper sizes in cartesian decimal units (0.1mm) and in polar units (0.035mm)

paper	cartesian units X	Y	polar units X	Y
A4	2100	2970	$1781	$213E
A	2150	2780	$1810	$1F1D
A3	2970	4200	$213E	$2F02
B	2780	4300	$1F1D	$3020

The table below shows *absolute cartesian co-ordinates* of paper edges in decimal units (polar mode has no absolute co-ordinates).

paper	x-edge	y-edge
A4	-500 to 1600	-500 to 2470
A	-500 to 1650	-500 to 2280
A3	-500 to 2470	-500 to 3700
B	-500 to 2280	-500 to 3800

All of the paper area can be used for plotting, provided the robot meets no obstructions. For best plot accuracy, keep the drive wheels entirely on the paper by limiting plot areas as follows (absolute cartesian co-ordinates, decimal units):

paper	X	Y
A4	210 to 890	210 to 1760
A	210 to 940	210 to 1570
A3	210 to 1760	210 to 2990
B	210 to 1570	210 to 3090

What to do if things go wrong

If nothing happens at all

- **Self-test routine** to confirm that Penman is working properly
- **Initialising Penman** – if fails, indicates that either cable or computer is faulty

If Penman beeps

Single beep

□ while running an application or tested program
The robot has met an obstacle while moving. If this happens during plotting, clear the obstacle and replot on a new sheet of paper.

□ while developing a program
A single beep probably means that you have sent Penman a command with a syntax error or an out-of-range parameter. If so attend to the program and re-send the command.

Series of beeps

Penman is still receiving commands from your computer after it has sent a 'handshake' to stop them temporarily. This may be due to faulty handshaking over the serial link. Check the cable – it may be loose, or you may be using the wrong cable type. Make sure that your computer serial port is configured to take notice of the handshake.

Unexpected output (gross errors, not minor mis-alignments)

This is probably due to mistaken operation of Penman or your computer. Start the plot again, making careful note of all the steps you take. If the problem persists, contact your supplier for advice.

AXIOM
CORPORATION
1014 GRISWOLD AVE.
SAN FERNANDO, CA 91340
(818) 365-9521